合肥工业大学大学计算机与信息学院

**《单片机课程设计》报告**

**（2022 -2023 学年第二学期）**

课程设计题目：单点采集温湿度显示并控制继电器姓名： 季宏鑫

学号： 2020213373

班级： 电子信息工程20-01班

指导老师： 蒋薇薇

时间： 2023-03-01

成绩：

## 1、设计目的

设计一个温湿度检测和控制的单片机系统，能够自动检测出当前情况的温度和湿度数值并且实时上传到onenet云平台进行数据存储和可视化显示。在使用过程中，用户可以自己设置温湿度系统的温度上下限和湿度上下限，当单片机检测用户设置的上下限时，会自动启动继电器带动工作电器对其进行降温操作。

## 2、设计任务

通过温湿度传感器DHT11采集温、湿度，能够通过显示单元显示出温、湿度，并控制继电器。

1. 开机液晶屏上显示姓名、学号3秒
2. 显示当前温、湿度和设定的温度、湿度上下限。

温度测量范围：

温度测量误差：

湿度测量范围：

湿度测量误差：

测量精度：

测量数值均为整数

1. 当温度或者湿度超过上下限时启动继电器带动风扇。
2. 可以通过矩阵键盘调节温度、湿度上下限。

可调控温度范围：

可调控湿度范围：

1. 温度、湿度上下限设置后断电不丢失。
2. 通过ESP8266模块连接WIFI上传数据至onenet物联网平台，并进行可视化展示。

上传云平台的网络IP地址：183.230.40.33

ESP8266设置为客户端模式：AT+CIPMODE=1

单片机与上位机进行通信，通过单片机上位机实时显示获取到的温湿度数据

1. 通过微信小程序在后台对温度、湿度信息进行监测。

微信小程序开发平台：微信开发者工具

## 3、原理框图及说明

1. 测量原理

根据设计任务，设计如下框图：

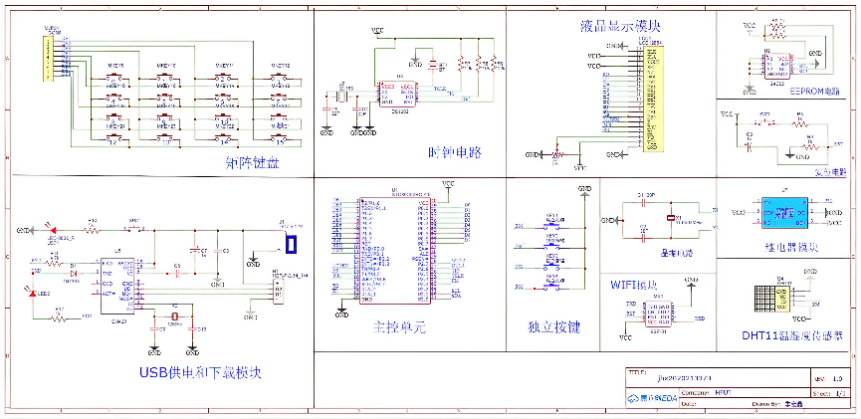


图1、原理框图

主控单元，主要负责从DHT11传感器中获取温度、湿度信息，获取矩阵键盘的设置温湿度信息，并将设置的温湿度信息和温湿度上下限显示在液晶显示模块上，并通过RX、TX串口发送数据到上位机以及ESP8266WIFI模块中去，便于向云平台传输数据。当温度或者湿度超过设定的温湿度阈值时，主控单元通过控制设置的继电器引脚电平带动直流电机工作。

电源部分，为单片机提供5V稳压电源和3.3V稳压电源。

矩阵键盘部分，其中K1~K9键盘用来设置温度湿度阈值，K13和K14按键可以用来设置单片机的显示界面和发送数据界面。

独立按键部分，其中K1~K4分别代表温度上限、温度下限、湿度上限、湿度下限四个部分的设置。

液晶显示模块，用来显示初始界面（姓名学号显示3s）、设置温度界面、显示温度界面、温度上下限、发送数据界面。

EEPROM电路，用来存储设置的温度湿度上下限信息，当突然断电时，保存信息，当下次通电时，读取上次设置的温湿度信息并且展示液晶显示模块上。

WIFI模块，用来接收DHT11温度传感器获取到的温湿度数据并通过AT指令发送给网址和指定的设备。

晶振电路以及时钟电路：晶振电路主要是用来提供一个固定频率的脉冲方便时钟电路计时；时钟电路主要是用来为第一个功能的3s钟进行计时，以及为后面的显示模块刷新率进行设置。

1. 使用说明

根据原理框图，设计电路图如附件1.

首先打开单片机电源，烧录编写好的程序

单片机显示模块会显示开机页面即姓名、学号信息3s。

3s之后会进入温度和湿度设置界面，通过独立键盘模块K1~K4依次选择温度下限、温度上限、湿度下限、湿度上限并通过矩阵键盘K1~K9进行按键设置。如果出现设置错误的情况只需要再通过独立键盘选择错误的设置参数并通过矩阵键盘重新设置即可。

当温湿度阈值设置完成后，按下矩阵键盘K13即可进入温湿度显示模块，该模块会显示当前的温湿度信息和设置的温湿度上下限。

按下矩阵键盘的K14即可把温湿度信息发送到OneNet云平台进行数据存储和可视化展示并实时同步到微信小程序平台进行后台监测。

连接上usb上位机即可实时显示获取的温湿度数据。

当温度和湿度超过设定的上下限时，启动继电器带动直流电机工作。

当整个系统断电后，再次打开单片机系统会自动保存之前设置的温度和湿度上下限。

## 4、主要电路说明、元件选择及参数计算

|  |  |
| --- | --- |
| **内部模块** | **数量*漫*** |
| 核心芯片STC89C52 | 1*漫* |
| 电容 | 若干个 |
| 电阻 | 若干个 |
| *漫*杜邦线 | 若干个 |
| USB转串口 | 1*漫* |
| 3.3V 5V供电电压 | 6 |
| 按键 | 4*漫* |
| 蜂鸣器 | 1 |
| 72M*漫*晶振*漫* | 1*漫* |
| 矩阵键盘模块 | 1 |
| 24C02芯片 | 1 |
| **外部模块** | **数量*漫*** |
| DHT11温湿度传感器 | 1 |
| ESP8266WIFI模块 | 1 |
| 直流电机 | 1 |
| 1路5V继电器 | 1 |

### 4.1复位与振荡电路

复位电路如图2，由C14和R9构成。由于51单片机为低电平复位，图示电路利用电容的特性，在刚刚外加5V电源时RST脚的电压为低。时间常数=RC=1000mS，满足芯片复位要求。由于电压为5V，可选择耐压6.3V的电解电容即可。

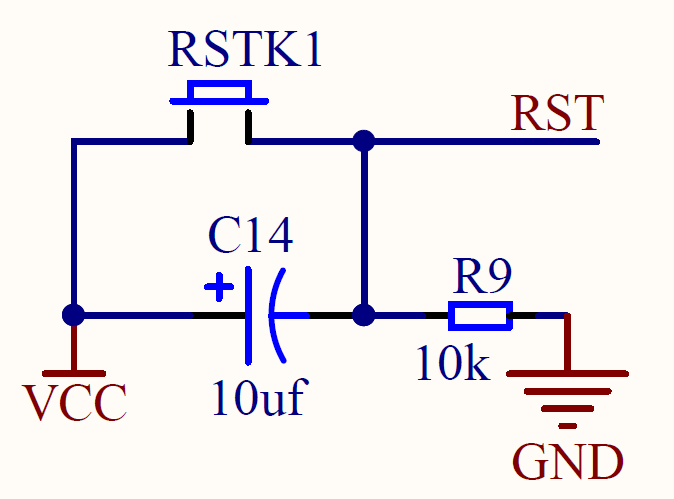


图4.1 复位电路

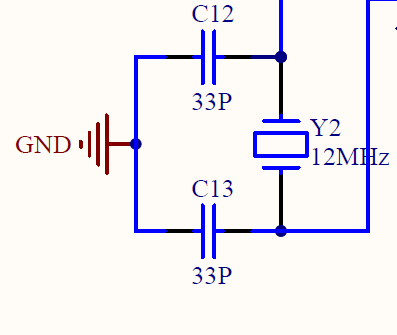


图4.1.1 振荡电路

振荡电路按照说明书的推荐，选择12MHz晶体，C7、C8选择33pF瓷片电容。

### 4.2 显示电路

经过大量的测试和调研，最终我的显示元件选择20引脚的12864液晶显示屏（128X64显示分辨率为128×64，内置8192个16\*16点汉字，和128个16\*8点ASCII字符集。利用该模块灵活的接口方式和简单、方便的操作指令。可以显示8×4行16×16点阵的汉字。也可完成图形显示），这里把各个引脚的功能以及连接方式做成了一个表格，以便直观地显示，通过51单片机上自带的可调电阻来调节12864的背光。

表4.2.1 lcd12864引脚功能以及接法

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| GND | +5V | V\_com | RS | R/W | EN | D0 | D1 | D2 | D3 |
| 模块的电源地 | 模块的电源正端 | LCD驱动电压输入端 | 并行的指令/数据选择信号 | 并行的读写选择信号 | 并行的使能信号 | 数据0 | 数据1 | 数据0 | 数据0 |
| GND | VCC | 电位器 | P2^6 | P2^5 | P2^7 | P0^0 | P0^1 | P0^2 | P0^3 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D4 | D5 | D6 | D7 | PSB | NC | RST | V\_OUT | BLA+ | BLA- |
| 数据4 | 数据5 | 数据6 | 数据7 | 并/串行接口选择 | 空脚 | 复位 低电平有效 | 空脚 | 背光源正极 | 背光源和负极 |
| P0^4 | P0^5 | P0^6 | P0^7 | P3^2 |  | VCC |  | VCC | GND |

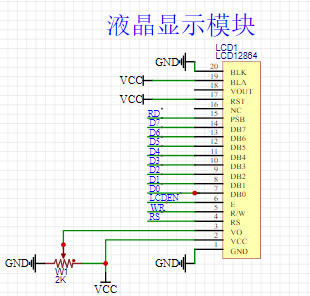


图4.2.1 显示电路

### 4.3按键电路

由于89C52的P3口内部有上拉电阻，为简化电路，即当K1没有按下时，P3.1=1，当K1按下时，P3.1=0。

矩阵键盘本质是使用8个io口来进行16个按键的控制读取，可以减小io口的使用，用4条I/O线作为行线，4条I/O线作为列线组成的键盘。在行线和列线的每个交叉点上，设置一个按键。而这样的按键中按键的个数是4 X 4个。第一行接到p17，第二行接到p16，第三行接到p15，第4行接到p14，第一列接到p13，第二列接到p12，第三列接到p11，第四列接到p10。先从P1口的高四位（四个行）输出高电平，低四位（四个列）输出低电平，假设有按键按下，从P1口的高四位读取键盘状态。判断高四位的四行哪一行变成了低电平，就知道是第几行，再从P1口的低四位（四个列）输出高电平，高四位（四个行）输出低电平，从P1口的低四位读取键盘状态。判断低四位的四列哪一行变成了低电平，就知道是第几列，将两次读取结果组合起来就可以得到当前按键的特征编码。

表4.3独立按键的连接电路

|  |  |  |  |
| --- | --- | --- | --- |
| K1 | K2 | K3 | K4 |
| P3.1 | P3.0 | P3.2 | P3.3 |

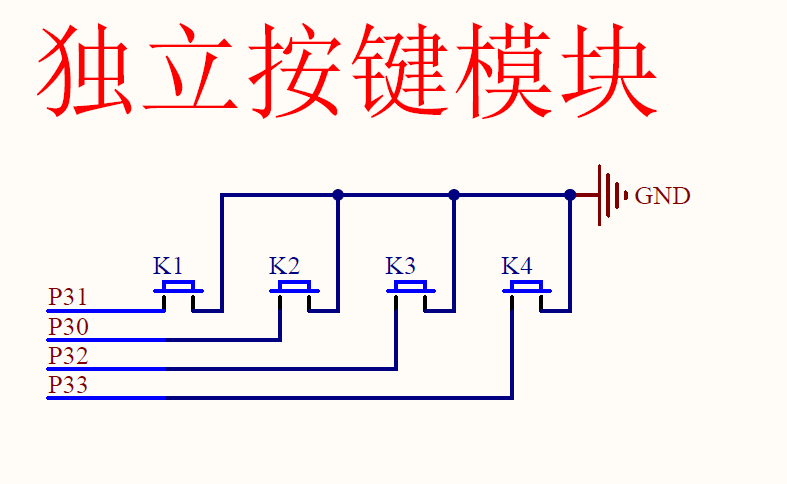
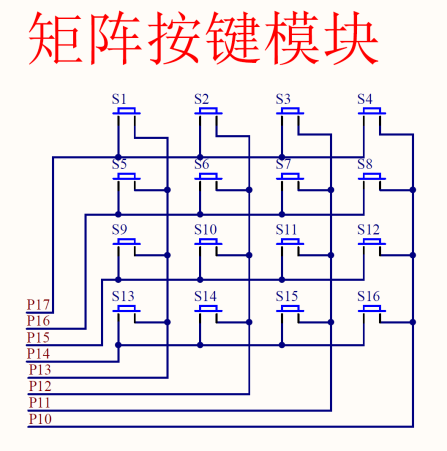


图4.3.1 独立按键电路

  
图4.3.2 矩阵按键电路

### 4.4 温度湿度检测电路

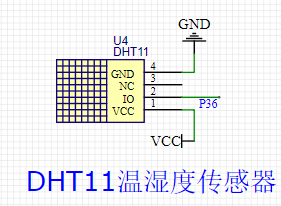


图4.4.1 温湿度传感器电路

经过大量的测试和我们所有的知识储备，最终我选择DHT11模块对温度，湿度进行检测并显示在12864显示器上。DHT的引脚图如下所示。

DHT11共有三个引脚，其中VCC，GND分别接入单片机电源模块上的VCC,GND。程序初始化后，单片机首先向DATA管脚发送一个低电平脉冲，如下图所示

表4.4.1温度模块引脚接法

|  |  |  |
| --- | --- | --- |
| VCC | GND | DATA |
| VCC | GND | P3^6 |

**（1）触发DHT11采集数据**

总线空闲状态为高电平,单片机把总线拉低等待DHT11响应,单片机把总线拉低必须大于18ms,保证DHT11能检测到起始信号。

当DHT11接收到单片机的开始信号后,等待单片机开始信号结束,然后发送80us低电平响应信号。

单片机发送开始信号结束后,延时等待20-40us后,切换为输入状态，等待DHT11的80us低电平信号结束，然后判断DHT11是否是否发出 80us 的高电平；如果是，即可开始采集数据。

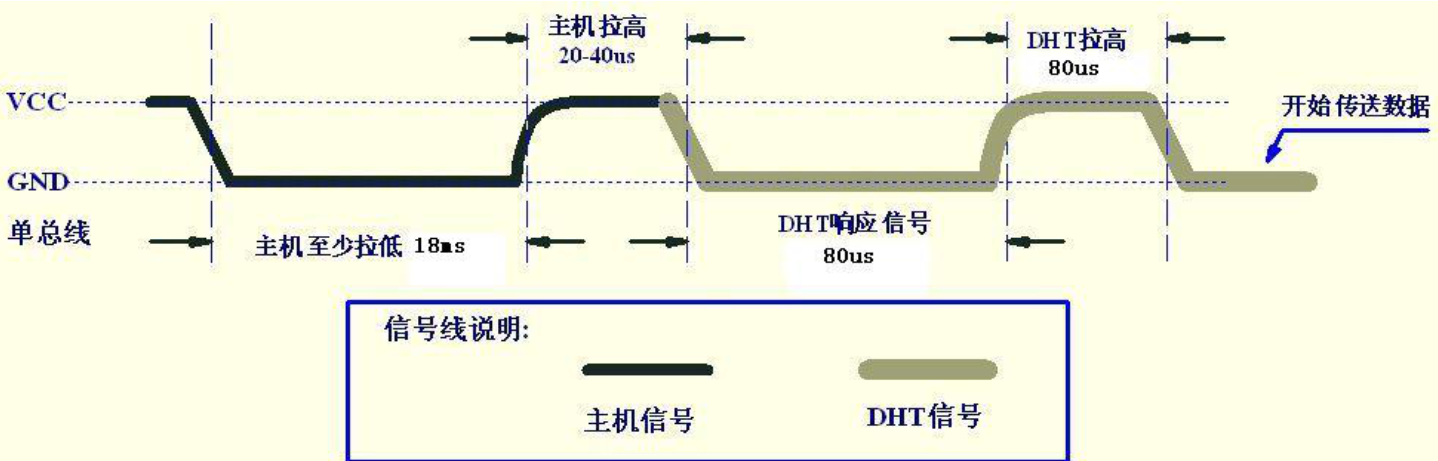


图4.4.2 采集数据的时序图

**(2)数字0和1的信号时序**

当DHT11输出数字0时， 单片机读取到的信号为50 us的低电平，之后为26-28 us的高电平。当DHT11输出数字1时， 单片机读取到的信号为50 us的低电平，之后为70 us的高电平。

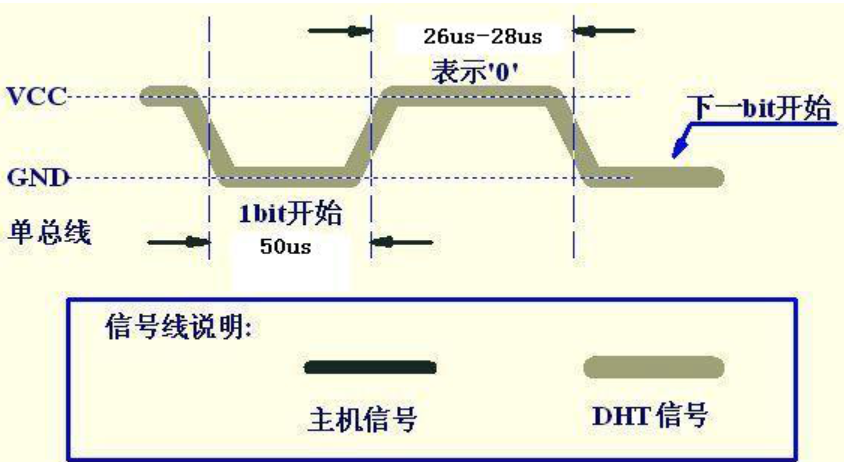
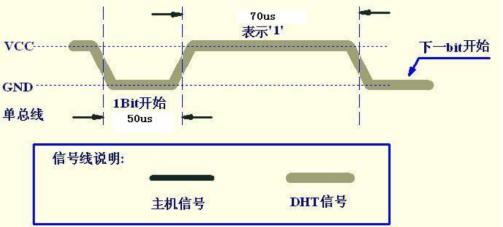
 

图4.4.3 数字0的时序图 图4.4.4 数字1的时序图

**(3)总时序图**

用户MCU发送一次开始信号后,DHT11从低功耗模式转换到高速模式,等待主机开始信号结束后,DHT11发送响应信号,送出40bit的数据,并触发一次信号采集,用户可选择读取部分数据。

从模式下,DHT11接收到开始信号触发一次温湿度采集,如果没有接收到主机发送开始信号,DHT11不会主动进行温湿度采集。当一次完整的采集数据后，DHT11会转换到低速模式。

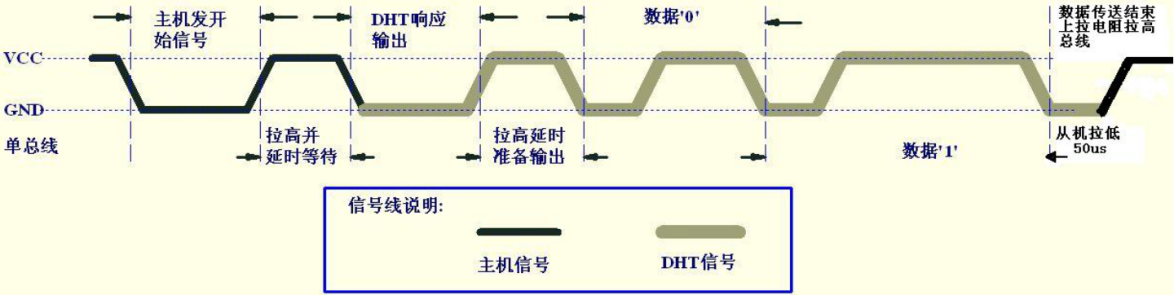


图4.4.5 总时序图

### 4.5 继电器模块

继电器模块如图4.5.1，当接好VCC,GND之后继电器模块会亮红灯，由于IN初始状态为低电平，三极管堵塞LED1灭。IN输入高电平后，三极管导通LED1亮同时K1被电磁铁吸引，开关从NC打到NO，从而实现电机电路的通断。

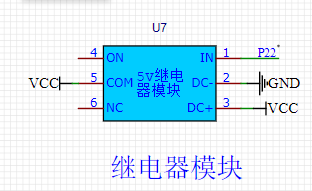


图4.5.1 继电器电路

当继电器线圈两端无电压，或者电压不够时，继电器的公共端（COM）与常闭端（NC）接通，当继电器线圈两端达到吸和的电压，继电器的公共端（COM）与常闭端（NC）接通

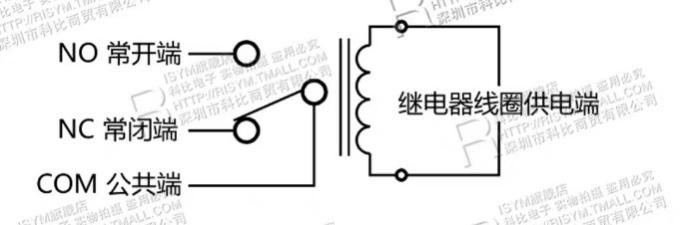


图4.5.2 继电器原理图

### 4.6 ESP8266WIFI模块

ESP8266是一款高性能的WIFI串口模块，内部集成[MCU](https://so.csdn.net/so/search?q=MCU&spm=1001.2101.3001.7020" \t "_blank)能实现单片机之间串口通信，是目前使用最广泛的一种WIFI模块之一。接线时只需要将RXD、TXD、GND、VCC这4个引脚，分别和USB转TTL模块的TXD、RXD、GND、VCC相连接。

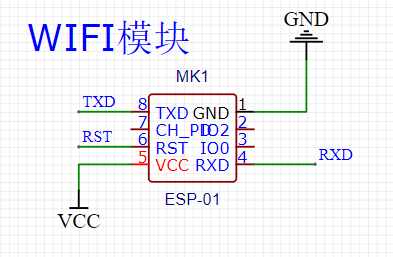


图4.6.2 ESP8266原理图

然后再通过USB转TTL模块和AT指令集通过以下步骤在串口助手中或者直接在单片机程序中对ESP8266模块进行配置。

1. AT——测试ESP8266是否工作完好
2. AT+CWMODE=1——设置ESP8266的工作模式为Station模式
3. AT+CWJAP=”ji”,”qqaazz112233”——进行WIFI连接（注意这里的手机热点或者外部热点只能是英文名称）
4. AT+CIPSTART=”TCP”,” 183.230.40.33”,”8080”——建立TCP连接，设置IP地址和PORT端口。
5. AT+CIPMODE=1——开启ESP8266的透传模式。
6. AT+CIPSEND——准备发送数据
7. POST /devices/1051580320/datapoints?type=3 HTTP/1.1 ——onenet的设备

api-key:1ZXQyfKnqL1bqEV0D0JJqbL7pmc= ——onenet的master key

Host:api.heclouds.com ——onenet的主机ip地址

Content-Length:21 ——传输的数据长度

{"temp":temp,"humi":humi} ——传输的数据



图4.6.2 ESP8266常用指令集

### 4.7 EEPROM电路

EEPROM电路的主要部分为AT24C02，原理图如下所示。

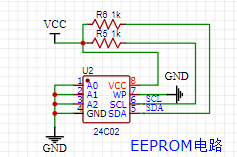


图4.7.1 EEPROM电路原理图

AT24C02 器件地址为 7 位，高 4 位固定为 1010，低 3 位由 A0/A1/A2 信号线的电平决定。 因为传输地址或数据是以字节为单位传送的，当传送地址时，器件地址占 7 位，还有最后一位（最低位 R/W）用来选择读写方向。  
 A0/A1/A2 默认连接到 GND，所以器件地址为1010000，即 0x50（未计算最低位）。如果要对芯片进行写操作时，R/W 即为 0，写器件地址即为 0XA0；如果要对芯片进行读操作时，R/W 即为 1，此时读器件地址为 0XA1。开发板上我们也将 WP 引脚直接接在 GND 上，此时芯片允许数据正常读写。

表4.7.1 AT24C02的芯片格式

|  |  |  |
| --- | --- | --- |
| **引脚号** | **引脚名称** | **功能说明** |
| **1** | A0 | 地址输入。A2、A1和IA0是器件地址输入引脚。 24C02/32/64使用A2、A1和AO输入引脚作为硬件地址，总线上可同时级联8个24co2/32/64器件(详见器件寻址)。 24C04使用A2和IA1输入引脚作为硬件地址，总线上可同时级联4个24C04器件,A0为空脚，可接地。 24C08使用A2输入引脚作为硬件地址，总线上可同时级联2个24CO8器件，AO和IA1为空脚、可接地。 24C16未使用器件地址引脚，总线上最多只可连接一个16K器件，A2、A1和AO为空脚，可接地。 |
| **2** | A1 |
| **3** | A2 |
| **5** | SDA | 串行地址和数据输入/输出。SDA是双向串行数据传输引脚，漏极开路，需外接上拉电阻到Vcc(典型值10kQ) . |
| **6** | SCL | 串行时钟输入。SCL同步数据传输，上升沿数据写入，下降沿数据读出 |
| **7** | WP | 写保护。WP引脚提供硬件数据保护。当WP接地时，允许数据正常读写操作:当WP接Vcc时，写保护,只读。 |
| **4** | GND | 地 |
| **8** | VCC | 正电源 |

表4.7.2 AT24C02的芯片格式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | A2 | A1 | A0 | R/W |

## 5、总电路设计

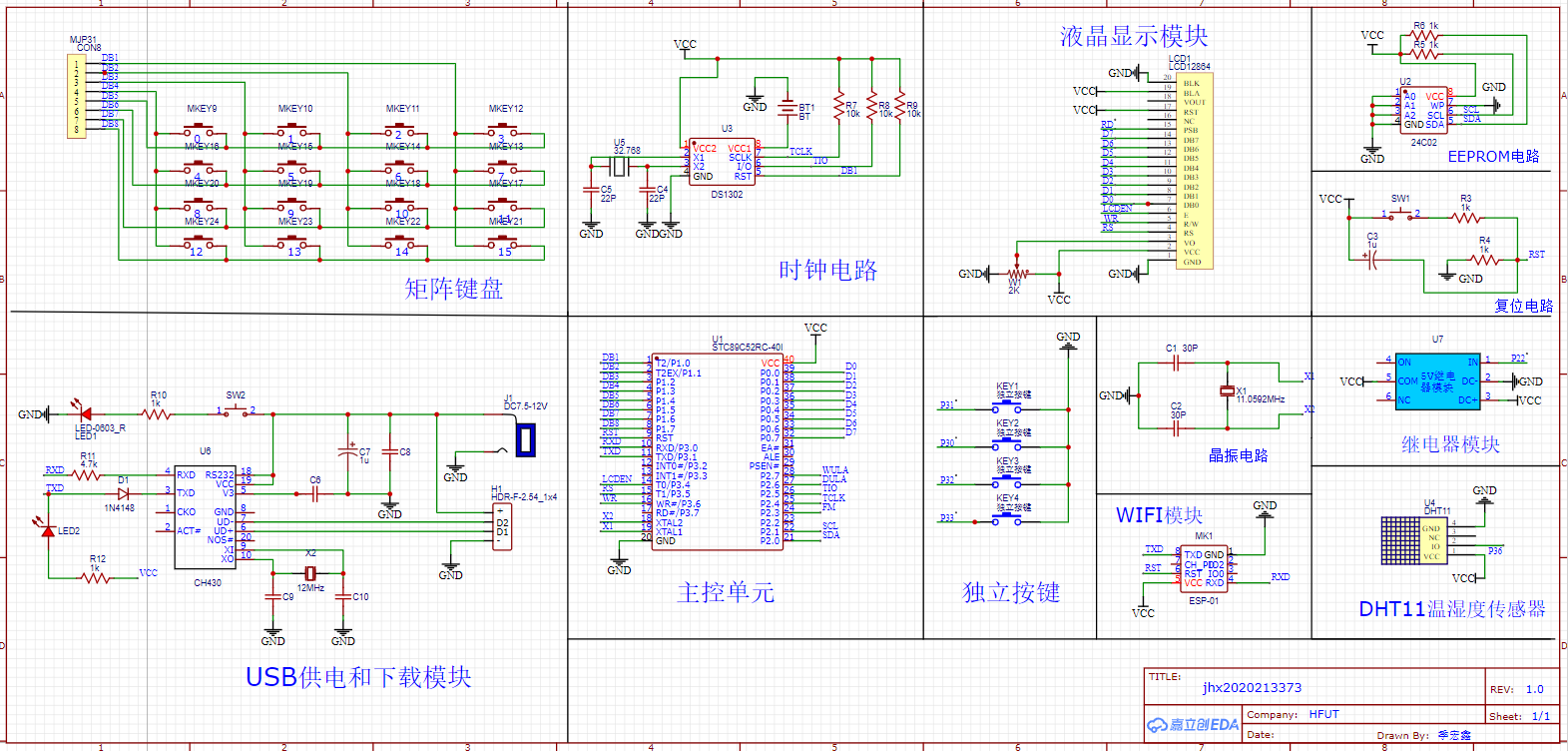


图5.1 总电路硬件连接图

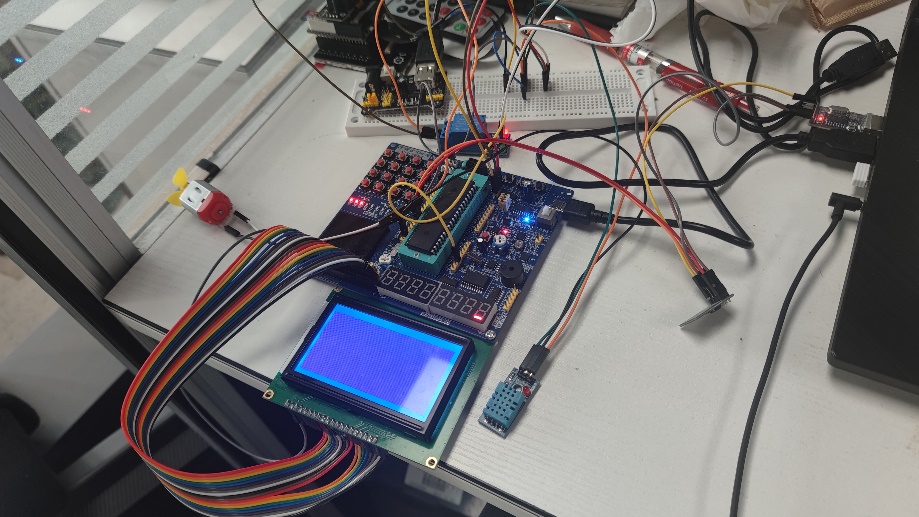


图5.2 总电路硬件实物图

## 6、软件设计思路及框图

### 6.1软件设计思路及主流程图

根据题目的设计要求以及我自己附加的功能，我采用定时器0和定时器1作为时钟数据源，定时器0每用来完成功能1姓名学号显示3s，定时器1用来对ESP8266和II2C进行初始化并实现发送数据、存储数据以及初始化数据线的功能。设置温湿度、显示温湿度信息和上下限、继电器以及直流电机的工作、矩阵键盘扫描、比较当前数据和设定阈值的差别、发送温湿度数据等全部放在主程序中实现。

温湿度获取部分设置数据缓冲区，DHT11传感器获取的温湿度信息直接送入缓冲区，由程序完成动态扫描。

**设计的全局变量：**

1. 温湿度数据缓冲区：uchar wendu[4]、uchar shidu[4]
2. 设置的温湿度上下限：uchar temtop[4]、uchar temlow[4]、uchar humtop[4]、uchar humlow[4]
3. 温湿度传感器接收数据：uchar rec\_dat[9]
4. 矩阵键盘的扫描值uchar key、uint flag
5. 独立键盘的扫描值：uchar func
6. ESP8266接收状态标记：uint UART\_RX\_STA
7. ESP8266接收缓冲：uchar UART\_RX\_BUF[UART\_REC\_LEN]

**设计的端口：**

1. **LCD端口**

sbit LCD\_RS = P2^6; //寄存器选择输入

sbit LCD\_RW = P2^5; //液晶读/写控制

sbit LCD\_EN = P2^7; //液晶使能控制

sbit LCD\_PSB = P3^2; //串/并方式控制

1. DHT11端口

sbit Data=P3^6;

1. 继电器端口

sbit wind=P2^1;

1. 蜂鸣器端口

sbit beep=P2^5;

**定义的端口：**

1. #define LCD\_data P0 ——LCD数据口
2. #define keyboard P1 ——定义矩阵键盘
3. #define UART\_REC\_LEN 10 ——定义ESP8266最大接收字节数

**设计函数如下：**

（1）主函数 main()

入口：无，出口：无。 一个大的循环，程序用不结束。

（2）ESP8266发送数据 void send\_data()

功能：对ESP8266进行云平台连接并通过ESP8266发送获取的DHT11温湿度传感器获取的数据。

入口：无，出口：无0

（3）初始化定时器0 void time1\_init(void)

功能：初始化定时器的工作模式，并对定时器0赋初值。

入口：无，出口：无。 改动当前显示位号。

（4）定时器的中断函数 void time0() interrupt 1

功能：通过设置count计数变量，每1mS中断一次，记录中断次数到3000次则初始化页面跳转至设置温度界面，并调整为合法的时间格式。

入口：无，出口：无。

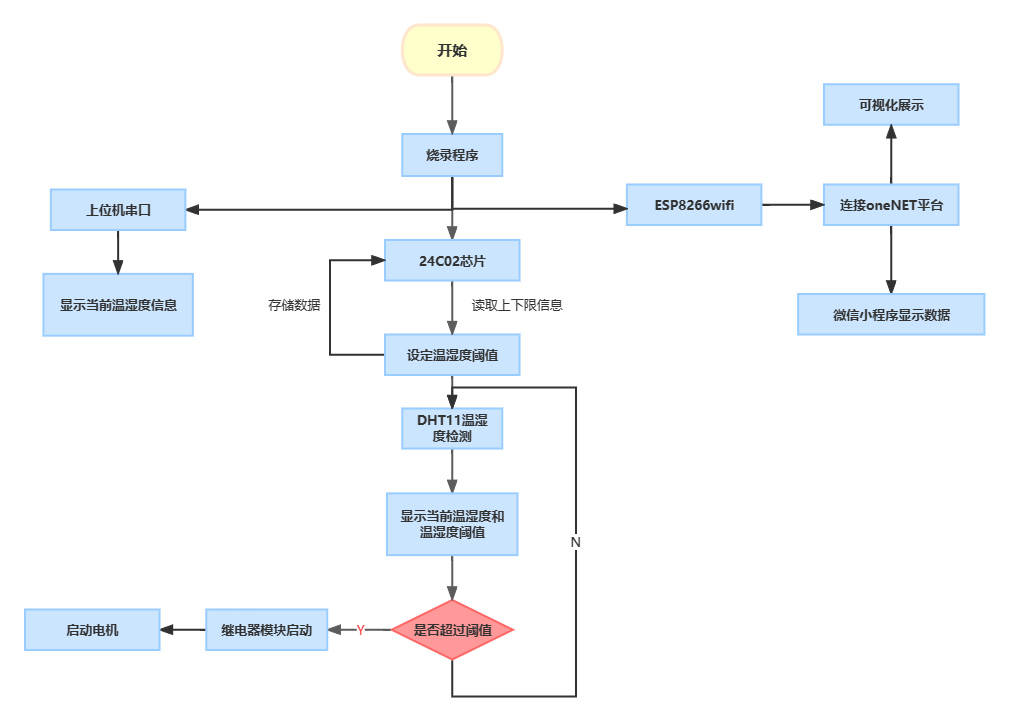


图6.1.1 主程序框图

### 6.2 延时模块

**设计函数如下：**

1. void delay\_ms(uint ms)

功能：设置延迟ms的函数

入口：延迟的时间，出口：无

1. void delay(int ms)

功能：设置延迟ms的函数

入口：延迟的ms时间 出口：无

1. void delay\_us(uchar n)

功能：设置延迟的us的函数

入口：延迟us的函数 出口：无

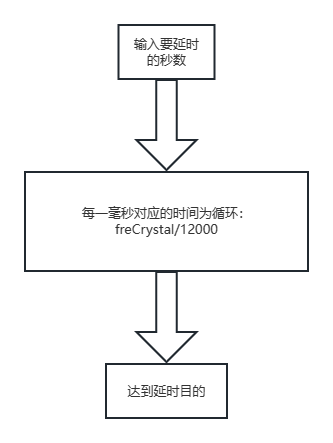


图6.2.1 主程序框图

### 6.3 LCD12864显示模块

**设计的端口值：**

sbit LCD\_RS = P2^6; //寄存器选择输入

sbit LCD\_RW = P2^5; //液晶读/写控制

sbit LCD\_EN = P2^7; //液晶使能控制

sbit LCD\_PSB = P3^2; //串/并方式控制

**定义的端口值：**

#define LCD\_data P0 //数据口

**设计函数如下：**

1. bit lcd\_busy()

功能：检查LCD忙状态

入口：无，出口：，lcd\_busy为1时，忙，等待。lcd-busy为0时,闲，可写指令与数据。

1. void lcd\_wcmd(uchar cmd)

功能：写指令数据到LCD

入口：LCD指令集 出口：无

1. void lcd\_wdat(uchar dat)

功能：写显示数据到LCD

入口：写入的数据 出口：无

1. lcd\_init()

功能：LCD初始化设定

入口：无 出口：无 调用函数：lcd\_wcmd,lcd\_busy()

1. void lcd\_showstring(uchar x,uchar y,uchar \*str)

功能：LCD根据设定的位置显示字符串

入口：设置的x轴和y轴，显示的字符串 出口：无 调用的函数：lcd\_wdat(uchar dat)

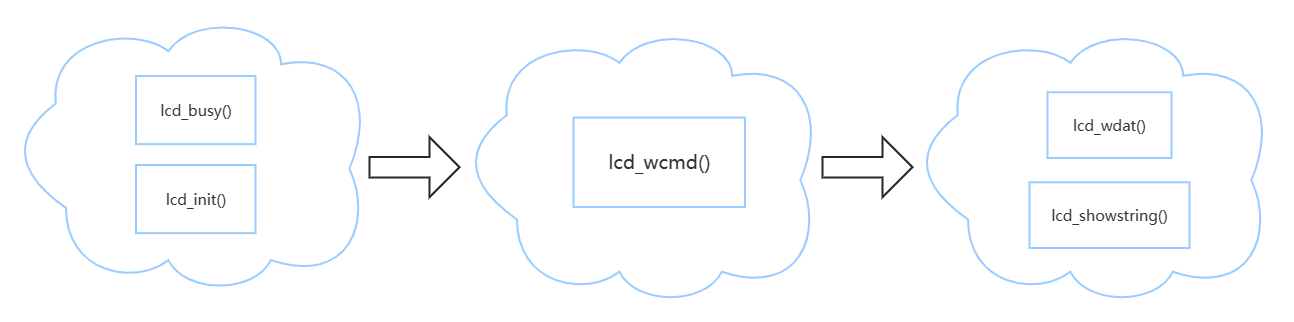


图6.3.1 lcd12864模块流程图

### 6.4 按键扫描模块

**定义的端口值：**

#define keyboard P1 ——设置键盘数据口

**设计函数如下：**

uchar Keypress()

功能：扫描哪个按键按下了

入口：无，出口：按下的按键

### 6.5 DHT11传感器数据获取模块

**设计的全局变量：**

uchar rec\_dat[9];

**定义的端口：**

sbit Data=P3^6;

**设计函数如下：**

1. DHT11\_start()

功能：初始化DHT11温湿度传感器模块

入口：无，出口：无

1. DHT11\_rec\_byte()

功能：DHT11接收获取到的字节数据

入口：无 出口：接收到的字节数据

1. DHT11\_receive()

功能：接收所有的温度湿度电平数据并进行转换

入口：无 出口：转换的温湿度数据

1. DHT11\_temget()

功能：把转换过的数据存进湿度和温度数组中去

入口：无 出口：无 调用函数：lcd\_wcmd,lcd\_busy()

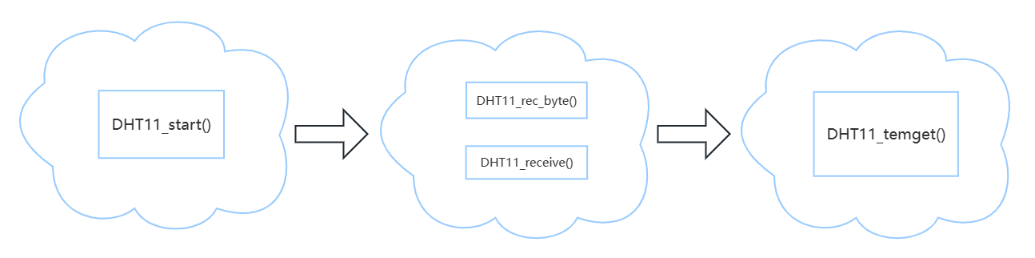


图6.5.1 DHT11温湿度传感器获取温度湿度

### 6.5 显示温湿度界面

**设计的全局变量：**

uchar number[]="123456789"; ——矩阵键盘的按下值

uchar temtop[4]; ——设置的温度上限值

uchar temlow[4]; ——设置的温度下限值

uchar humtop[4]; ——设置的湿度上限值

uchar humlow[4]; ——设置的湿度下限值

uchar wendu[4]; ——温度数据缓存值

uchar shidu[4]; ——湿度数据缓存值

uint ttop,tlow,htop,hlow,tnow,hnow; ——把温度湿度上下限数组缓存值转化为十进制数

**定义的端口：**

sbit beep=P2^5; ——蜂鸣器引脚

sbit wind=P2^2; ——继电器引脚

**设计函数如下：**

1. showtem()

功能：显示当前温度和湿度

入口：无，出口：无

1. settem()

功能：设置温湿度上下限

入口：无 出口：无

### 6.6 ESP8266WIFI发送模块

**设计函数如下：**

1. UART\_Init(void)

功能：初始化UART协议和定时器1

入口：无 出口：无

1. UART\_SendData(uchar dat)

功能：发送一个字符数据

入口：字符数据 出口：无

1. UART\_SendString(uchar \*pbuf)

功能：发送一个字符串数据

入口：字符串数据 出口：无

1. ESP8266\_SendCmd(uchar \*pbuf)

功能：向ESP8266发送AT指令集

入口：AT指令，字符串格式，出口：无

1. ESP8266\_SendData(uchar \*pbuf)

功能：ESP8266 WIFI发送数据到APP

入口：数据 出口：无

1. ESP8266\_ModeInit(void)

功能：ESP8266 WIFI模块工作模式初始化

入口：无 出口：无

1. ESP8266\_SendData(uchar \*pbuf)

功能：ESP8266 WIFI发送数据到APP

入口：数据 出口：无

1. wifi\_control\_init(void)

功能：WIFI控制的初始化

入口：无 出口：无

### 6.7 EEPROM断电不丢失模块

**定义的端口：**

sbit sda = P2 ^ 0; //I2C总线数据线

sbit scl = P2 ^ 1; //I2C总线时钟线

**设计函数如下：**

1. c02\_init()

功能：初始化24C02芯片

入口：无，出口：无

1. start()

功能：启动II2C总线

入口：无 出口：无

1. stop()

功能：停止II2C总线

入口：无 出口：无

1. writebyte(unsigned char j)

功能：向数据线中写入一个字节

入口：一个字符数据 出口：无

1. readbyte()

功能：读取数据线中的一个字节

入口：无 出口：无

1. read24c02(unsigned char address)

功能：从24c02的地址address中读取一个字节数据

入口：24C02的地址 出口：无

1. write24c02(unsigned char address, unsigned char info)

功能：向24c02的address地址中写入一字节数据info

入口：地址，字符数据 出口：无

1. save\_data()

功能：保存温度和湿度数据

入口：无 出口：无

1. read\_data()

功能：读取保存的温度和湿度数据

入口：无 出口：无

## 7、结果及测试分析

### 7.1硬件电路

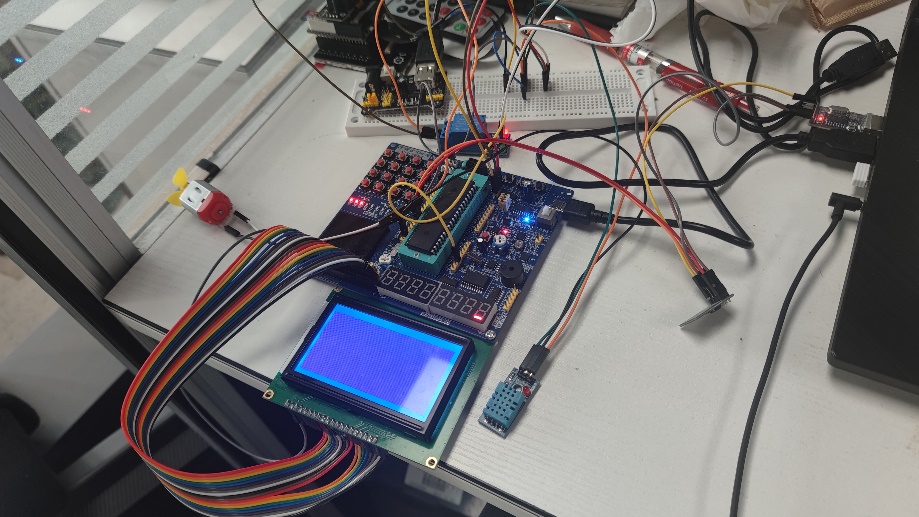


图7.1 硬件实物图

**说明：**

硬件部分的连接过程在原理图中都有具体列出。

对于电源模块，由于51单片机自带的电源输出端口只有一个5V和3.3V不能够满足我们众多外加模块的供电需求，所以我这里采用面包板模拟拓展5V电源和3.3V电源。

在连接实物图时由于51单片机自带的12864端口会遮挡住引脚的连接，所以在这里我采用杜邦线把51单片机端口与12864端口相连接。DHT11温湿度传感器直接连接我们配置好的引脚并连接电源即可。继电器模块由于我们采用的是电平触发模式，所以需要把继电器模块的IN端口连接至配置好的端口并接上电源即可，COM端口接地、NC端口接电源，NO端口悬空，直流电机分别于GND和NC端口相连从而达到继电器控制直流电机的效果。ESP8266WIFI模块的RX和TX分别于单片机的TX和RX相连并且连接上3.3V电源和GND就可以达到上传数据至上位机和云平台的效果。

### 7.2测试记录及结果分析

### 7.2.1 测试记录

1. 首先通过usb线把单片机和电脑相连烧录程序，烧录完程序将自动进入初始界面，初始界面如下图所示。

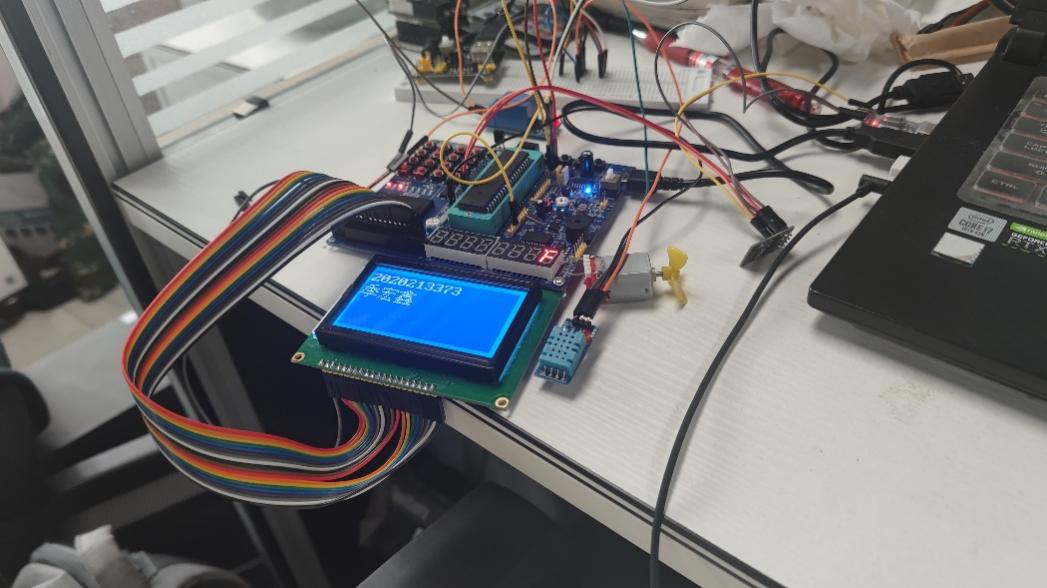


图7.2.1 初始界面图

1. 当显示姓名学号3s过后将会自动跳转至设置温度和湿度阈值的界面，然后通过独立键盘K1~K4分别设置温度上限、温度下限、湿度上限、湿度下限，通过矩阵键盘K1~K9设置温湿度阈值的具体数值

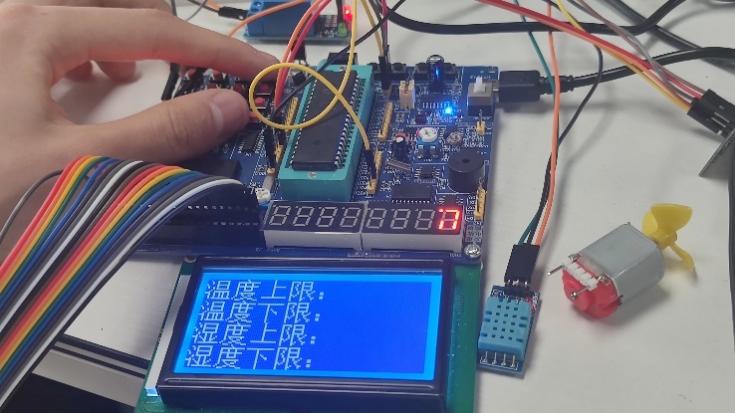


图7.2.2 设置温湿度阈值

1. 当设置好温湿度阈值时，点击矩阵键盘的K13即可进入温湿度显示页面，该页面可以显示温湿度的当前值和温湿度上下限值，显示界面如下图所示。



图7.2.3 温湿度显示界面

1. 当温度和湿度超过设定的阈值时，继电器会启动，继电器的指示灯变为绿灯。



图7.2.4 继电器启动

1. 当按下矩阵键盘的K14键即可把单片机获取的温湿度数据发送至onenet云平台进行存储，我还绘制了一个可视化平台用来实时展示单片机获取的数据。数据存储以及可视化展示如下图所示。

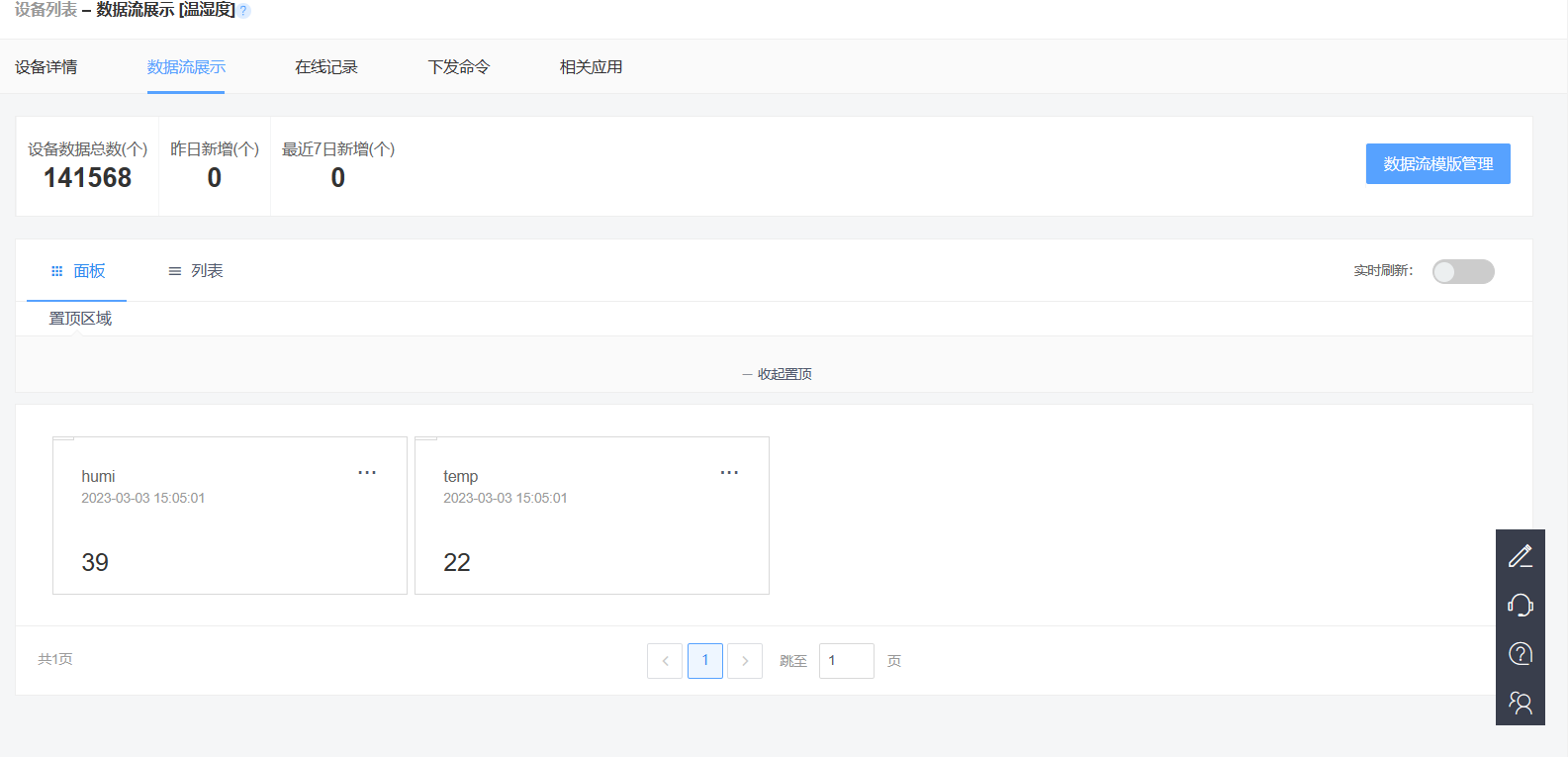


图7.2.5 云平台数据存储

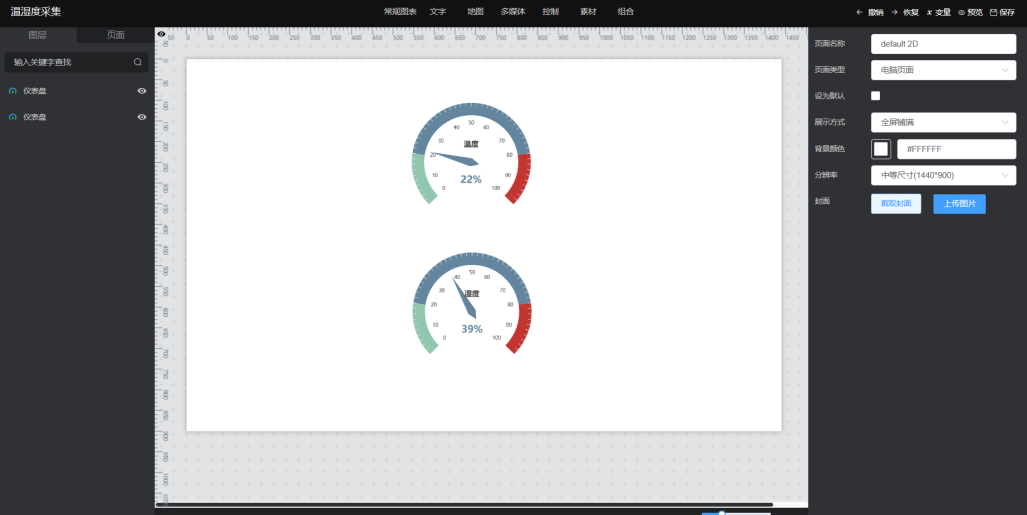


图7.2.6 可视化数据展示

1. 我还制作了配套的微信小程序方便客户在手机端实时对温湿度的情况进行掌控和调配，小程序的页面如下图所示。

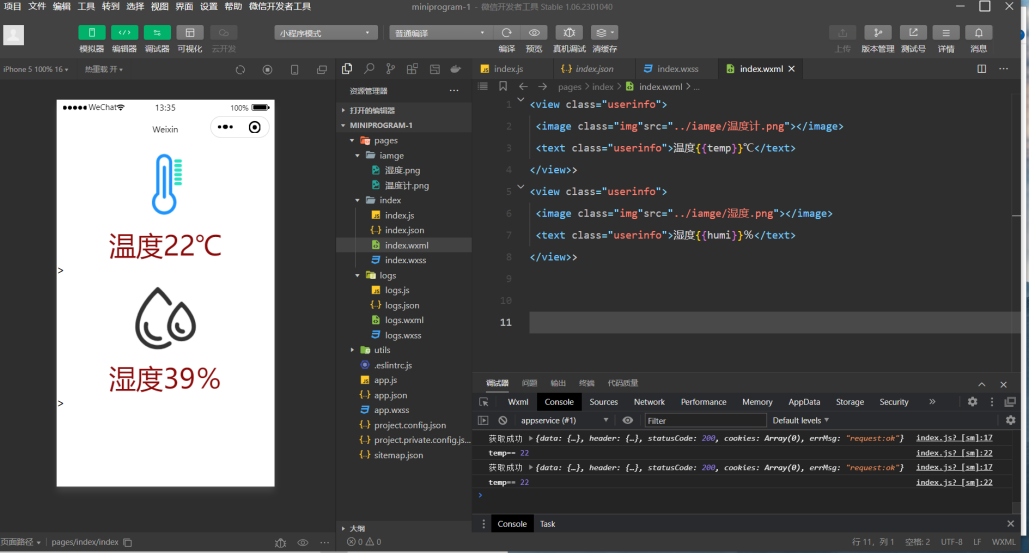


图7.2.7 微信小程序后台显示

1. 单片机与上位机通信，能够通过单片机实时把数据发送给电脑，电脑显示的数据如下图所示。

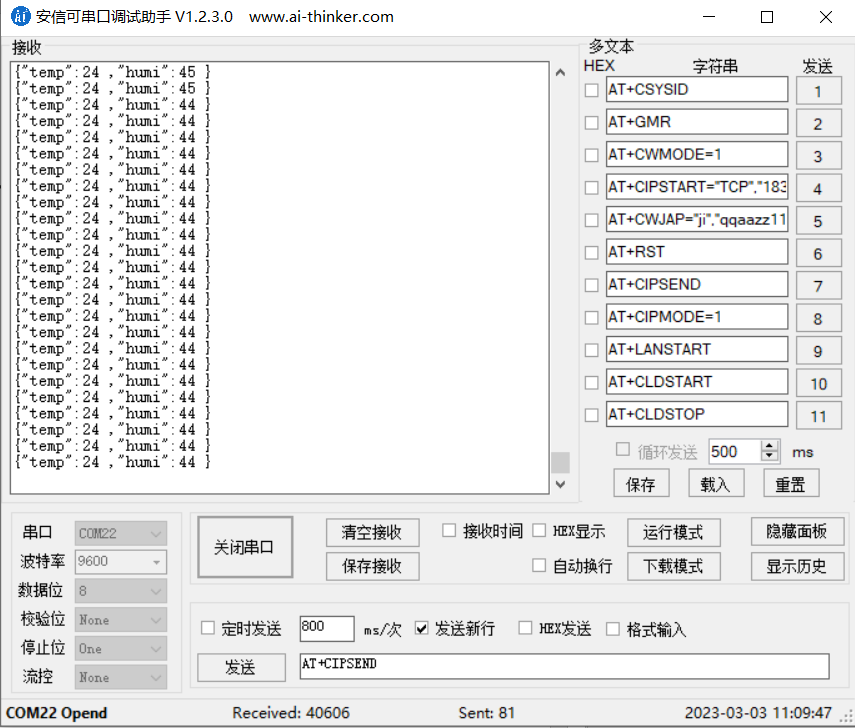


图7.2.8 微信小程序后台显示

1. 当断开单片机电源再接通单片机电源时我们设置的单片机上下限值会被保存，效果如下图所示。



图7.2.9 断电不丢失

### 7.2.2 结果分析

制作完成后，进行如下测试，结果合格。

|  |  |  |
| --- | --- | --- |
| 功能要求 | 功能描述 | 测试结果 |
| （1）显示姓名学号3s | 在LCD12864的显示屏上显示姓名学号3s | 结果一致，计时十分准确 |
| （2） 通过矩阵键盘设置温湿度上下限 | 通过独立键盘选择需要设置的属性，通过矩阵键盘设置具体的值 | 结果一致 |
| （3）显示当前温湿度和设定的温湿度上下限 | 通过LCD12864的显示屏上显示温度、湿度数据和设定的温湿度上下限 | 结果一致 |
| （4）当温度或者湿度超过设定的上下限时启动继电器带动风扇 | 当我们提高室温或者湿度超过设定的阈值会启动继电器 | 结果一致 |
| （5）温度、湿度上下限设置后断电不丢失 | 我们断开电源再接入电源时，我们设定的温湿度阈值不会丢失 | 结果一致 |
| （6）通过ESP8266连接WIFI上传数据至云平台 | 通过ESP8266连接WIFI并通过指定的AT指令集发送数据到onenet的指定设备 | 结果一致 |
| （7）单片机与上位机进行通信 | 通过单片机上位机实时显示获取到的温湿度数据 | 结果一致 |
| （8）通过微信小程序进行后台监测数据有无异常 | 开发微信小程序，获取单片机的温湿度值并进行显示和警告 | 结果一致 |

## 体会与感想

持续两周的单片机课程设计落下帷幕。我选的设计题目是单点采集温湿度显示并控制继电器，这道题看上去比较简单，但是这是我们日常生活中不可避免会碰到的控制系统，并且为了考虑到用户的具体体验，这其中其实有很多我们需要拓展的模块

在这次课程设计中我碰到了许多以前没用使用过的模块，比如DHT11模块、LCD12864模块、ESP8266WIFI模块等，对这些模块的不了解导致我坏了很长时间去应用好这些模块，经过网上不断地重复找资料，最后终于通过DHT11成功的利用了DHT11测出了室内的温度并在LCD12864显示屏上面显示出来而且还能够实现把温湿度数据通过esp8266发送给云平台进行数据存储。

在这次实验过程中我也碰到了很多的问题，比如ESP8266wifi模块在正常的3.3V电压下并不能够很好的工作，导致数据并不能发送出去，在查阅了很多的资料后我发现单片机本身自带的usb转ttl模块虽然有着3.3V的电压，但是实际的输出电压并没有达到3.3V而esp8266wifi对电源的要求较为严苛。所以在后期设计的过程中我使用了5V电源作为esp8266wifi模块的供电，最后完美的解决了这个问题。

由于我是第一次使用12864显示器，所以在调整显示模块的时候也遇到了很多以前使用1602没有遇到过的问题，比如有的时候12864会自动产生乱码等错误、会自动产生我没有编写的字符等等，最后通过询问老师和查找资料后我发现是由于我的12864显示模块的串口与单片机自带的红外模块串口发生了冲突导致有的时候会发生乱码的情况，最终我通过调整单片机的串口分配成功地解决了这个问题。

在完成单片机课程设计后，我发现自己还有许多不足，所学到的知识还远远不够，以至于还有一些功能不能被动完成。但透过学习这一次实践，增强了我的动手潜力，提高和巩固了单片机方面的知识，特别是软件方面。让我认识到把理论应用到实践中去是多么重要。

## 《单片机系统设计》评分表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **课程目标** | **指标点** | **评价观察点** | **评价方式** | | **得分（百分制）** |
| **依据/评价人** | **权重*wj*** |
| 1 | (CO1) 能通过查找文献及相关参考资料，对单片机系统设计任务进行需求分析和关键问题分析，确定解决方案和技术路线。 | GR3.1 | 过程中检查能否通过查找文献及相关参考资料，对设计任务进行需求分析、关键问题分析、确定解决方案和技术路线。 | 实际操作/教师 | 0.2 |  |
| 2 | (CO2) 能综合经济、健康、安全、法律、文化、环境等因素，选择合适的软硬件开发平台，设计、整合相关程序，能对现有方案、方法进行改进，并能通过调试、运行，验证方案的有效性、先进性。 | GR3.3 | 能否利用相关语言进行单片机系统的编程和设计实现；设计过程中态度及完成质量；验收过程中演示是否成功，能否正确回答老师的提问，并准确描述发现问题的过程和解决问题的办法。 | 验收/教师 | 0.3 |  |
| 设计报告的格式规范程度，是否图文并茂，语言规范及流畅程度；实验数据是否充分、结果是否正确，分析、对比是否充分，结论是否正确；是否提出了自己的独到见解。 | 设计报告/教师 | 0.3 |  |
| 3 | (CO3) 能够在系统开发及设计的过程中，认识到自主学习的必要性，发展终身学习的意识。 | GR12.1 | 是否能够通过查找资料自主学习，并且在完成设计要求的基础上对设计内容进行拓展，是否具有自主学习、终身学习的意识。 | 验收/教师 | 0.1 |  |
| 设计报告/教师 | 0.1 |  |
| 评分方法：评价人根据观察点要求对各项课程目标的完成情况进行评分（百分制）。  评分标准：1）观察点任务完成、能力达成，且表现突出：评为优秀，得分范围为85~100；  2）观察点任务完成、能力达成，表现良好：评为良，得分范围为75~84.9；  3）观察点任务完成、能力达成一般，表现较好：评为中，得分范围为66~74.9；  4）观察点任务基本完成、能力基本达成，表现一般：评为及格，得分范围为60~65.9；  5）观察点任务未完成、能力未达成，表现差：评为不及格，得分范围为0-59； | | | | | | |
| 总分： | | | | | | |